STUDY OF ONE- AND TWO-DIMENSIONAL FILTERING AND DECONVOLUTION ALGORITHMS FOR A STREAMING ARRAY COMPUTER

NASA GRANT NO. NSG 1648

FINAL REPORT

Appendix 4

Dr. George E. Ioup, Principal Investigator
Department of Physics
University of New Orleans
New Orleans, LA 70148

COORDINATE AXES, LOCATION OF ORIGIN, AND REDUNDANCY FOR THE ONE AND TWO DIMENSIONAL DISCRETE FOURIER TRANSFORM FOR COMPLEX AND REAL DATA

George E. Ioup and Juliette W. Ioup
Department of Physics and
Geophysical Research Laboratory
University of New Orleans, New Orleans, LA 70148

For continuous Fourier transforms, the shift theorem will locate the origin anywhere. If

$$\mathcal{F}\{f(x)\} = F(s) \quad ,$$

then

$$\mathcal{F}\{f(x-a)\} = \exp(-i2\pi as)\, F(s)$$

for forward transforms. For inverse transforms,

$$\mathcal{F}^{-1}\{F(s-a)\} = \exp(+i2\pi as)\, f(x) \quad .$$

The discrete Fourier transform (DFT) has special considerations: (1) there are only a finite number of sample points, and (2) there is replication in both domains.

COMPLEX ONE-DIMENSIONAL DATA

Consider a function domain array a_X, where x gives the coordinate location of each of the complex elements of a_X. The transform of a_X is given by the complex array A_J. Here J specifies the frequency number defined by $J\Delta s = s$. For $\Delta x = 1$, Δs is numerically equal to 1/N, with N the number of points in a_X. Its units are the reciprocal units of x. In general

$$\Delta s = 1/(N\Delta x)$$

Data arrays often have no particular origin associated with their elements. Those that do are commonly one of two forms:

Form 0a:

a_0 a_1 a_2 ... a_{N-2} a_{N-1}

which has N points, and

Form 1a:

a_{-M} a_{-M+1} ... a_{-2} a_{-1} a_0 a_1 a_2 ... a_{M-1} a_M

which has (2M+1) points. The standard convention for the DFT is

$$A_J = \sum_{k=0}^{N-1} a_k \exp(-i2\pi kJ/N) \quad ,$$

where k gives the position of a_k as measured from the left of the array (with the first element having k = 0) as stored by existing fast Fourier transform (FFT) programs. The form of the exponent is such that k = 0 is assumed to be the label of the origin element. Data in Form 0a can be directly transformed, but for data in Form 1a, something must be done to have the transform correctly represent the data with their given origin. As an alternative to having different FFT algorithms for different data origins, the data may be modified to use existing codes.

The required modification of the data is a rearrangement as follows:

Form 3a:

a_0 a_1 a_2 ... a_{M-1} a_M a_{-M+1} ... a_{-2} a_{-1}

for N = 2M+1, or

Form 4a:

a_0 a_1 a_2 ... a_{M-1} $a_{M,-M}$ a_{-M+1} ... a_{-2} a_{-1}

for $N = 2M$, with $a_{M,-M} = a_M + a_{-M}$. N is the number of points to be transformed.

Because the DFT corresponds to replicated data, the above modifications describe a shift of the data to the left by M elements, or of the origin to the right by M elements for N even, or by M+1 for N odd. If N is even, another form is needed to correspond to Form 1a. The first element of Form 1a is replaced by $a_{M,-M}$ and the last element dropped to give Form 2a.

Form 2a:

$a_{M,-M}$ a_{-M+1} ... a_{-2} a_{-1} a_0 a_1 a_2 ... a_{M-1} .

Another approach consists of taking the DFT of the unmodified data directly. The DFT definition is then used in its standard form and the DFT is corrected, rather than rearranging the data before the DFT is calculated. The DFT of the unshifted data can be changed into that of the shifted data by a phase multiplication determined by the Shift Theorem (Bracewell, 1978). Since the origin is assumed to be at the left-most element for the standard DFT algorithms, the a_x array must be shifted to the left so that a_0 will fall on the origin. For the forward DFT (-i transform), the phase multiplier corresponding to this shift will be $\exp(+i2\pi sM\Delta x)$. The multiplication of the DFT of data in Form 1a or 2a by this factor will result in a DFT corresponding to an origin correctly located at a_0.

The transform of sampled data is replicated, which means

that the DFT is periodically repeated. The period is $1/\Delta x$, which is often assumed to be 1. Since a periodic function is completely specified by one period (called an "island" by Bracewell, or principal interval, or base band), the discrete transform is given as one period of the periodic result. The base island of the transform, centered on the origin, is the part of the transform generally used for conceptual manipulations. Working with this island has the advantage that any frequency-dependent filtering will be simplified since the s values (frequency values) are simply associated with the elements of the transform array:

Form 1A:

	A_{-M}	A_{-M+1} ...	A_{-2}	A_{-1}	A_0	A_1	A_2 ...	A_{M-1}	A_M
s=	$-M\Delta s$	$(-M+1)\Delta s$...	$-2\Delta s$	$-\Delta s$	0	Δs	$2\Delta s$...	$(M-1)\Delta s$	$M\Delta s$

for N odd. Unfortunately, the convention for the DFT is not the above arrangement, but is

Form 3A:

	A_0	A_1	A_2 ...	A_{M-1}	A_M	A_{-M+1} ...	A_{-2}	A_{-1}
s=	0	Δs	$2\Delta s$...	$(M-1)\Delta s$	$M\Delta s$	$(-M+1)\Delta s$...	$-2\Delta s$	$-\Delta s$

for $N = 2M+1$, or

Form 4A:

	A_0	A_1	A_2 ...	A_{M-1}	$A_{M,-M}$	A_{-M+1} ...	A_{-2}	A_{-1}
s=	0	Δs	$2\Delta s$...	$(M-1)\Delta s$	$-M\Delta s$	$(-M+1)\Delta s$...	$-2\Delta s$	$-\Delta s$

for $N = 2M$, with $A_{M,-M} \equiv A_M + A_{-M}$. This arrangement may be thought of as consisting of the origin and right-hand half of the base island and the left-hand half of the first replication to the right. When filtering the transform in this form, care must be taken to associate the correct s value with each element. As an alternative to working with this DFT result,

the elements of the DFT may be rearranged into Form 1A for N odd or into

Form 2A:

$A_{M,-M}$ A_{-M+1} ... A_{-2} A_{-1} A_0 A_1 A_2 ... A_{M-1}

for N even. This may be accomplished by rearranging the DFT elements after they are obtained, or by phase multiplying the data before the transform is taken. Again the shift theorem is used to determine the phase multiplication. The phase multiplication for each a_x will be $\exp(-i2\pi M(\Delta s)x)$ if the -i transform is used for the DFT.

In the transform domain, the phase multiplication needed to correct for Form 2a with N even (M = N/2) is

$$e^{i2\pi Ms\Delta x} = e^{i2\pi(N/2)J\Delta s\Delta x} = e^{i2\pi(N/2)J\Delta x/(N\Delta x)} .$$

This becomes

$$e^{i\pi J} = (-1)^J .$$

Care must be taken that j measure the displacement in terms of positive or negative position number of the elements <u>from the s origin</u> of the transform. For $(-1)^J$, positive or negative makes no difference, but for other cases it will be important. The phase multiplier for N odd also simplifies, although not as much. This phase multiplication corrects for a data origin given by Form 2a.

For the function domain, the phase multiplication needed to have the transform be in Form 2A, N even, is

$$e^{-i2\pi(M\Delta s)k\Delta x} = e^{-i2\pi(N/2)(1/N\Delta x)k\Delta x} = e^{-i\pi k} = (-1)^k .$$

Again k measures the positive or negative displacement <u>relative to the x origin</u>. The simplification for N odd is similar but

not as great. This phase multiplication causes the transform to have Form 2A rather than Form 4A.

For Form 4A or 2A, for filtering purposes the element $A_{M,-M}$ is associated with $s = +M\Delta s$ and $s = -M\Delta s$ since $A_{M,-M} = A_M + A_{-M}$. If the filter is H(s), then the filtered result at this location should be

$$\frac{H(M\Delta s)A_{M,-M}}{2} + \frac{H(-M\Delta s)A_{M,-M}}{2}$$

if the a_X elements are real, since the transform is Hermitian (conjugate even or symmetric, Bracewell, 1978). For complex a_X, no rule can be given. Setting the filtered result to zero or to the result which holds for real data are two possible approaches.

TWO-DIMENSIONAL DATA

Consider Form 0a. Notice that the $a_{0,0}$ element is in the upper left corner, and subscripts increase down and to the right. This arrangement is chosen for two reasons; first, this is the usual way matrix elements are subscripted, and second, this is the usual order in which image data are stored in the computer. The subscripts can be used to denote the x and y axes of the two-dimensional transform. There are two choices: $a_{x,y}$: the x axis is vertically down and the y axis horizontal to the right, with the z axis out of the paper; or $a_{y,x}$: the x axis horizontal to the right and the y axis vertically down, with the z-axis into the paper. The latter convention is chosen here because the x and y axes are generally taken to be horizontal and vertical, respectively, in two-dimensional

coordinate systems. Note that the limits on x and y are $0 \le x \le N_2-1$ or $-M_2 \le x \le M_2$ and $0 \le y \le N_1-1$ or $-M_1 \le y \le M_1$. Forms 0a or 3a are assumed by standard FFT algorithms. In the notation of Bracewell the exponential in the two-dimensional Fourier transform is given by $\exp(-2\pi i(xu+yv))$. If x is chosen to be horizontal, then the u axis is horizontal; and if y is chosen to be vertical, then the v axis is vertical.

Form 0a:

$$\begin{bmatrix} a_{0,0} & a_{0,1} & a_{0,2} & \cdots & a_{0,N2-1} \\ a_{1,0} & a_{1,1} & a_{1,2} & \cdots & a_{1,N2-1} \\ a_{2,0} & a_{2,1} & a_{2,2} & \cdots & a_{2,N2-1} \\ \vdots & & & & \vdots \\ a_{N1-2,0} & a_{N1-2,1} & a_{N1-2,2} & \cdots & a_{N1-2,N2-1} \\ a_{N1-1,0} & a_{N1-1,1} & a_{N1-1,2} & \cdots & a_{N1-1,N2-1} \end{bmatrix}$$

Form 1a: M_1, M_2 odd

$$\begin{bmatrix} a_{-M1,-M2} & a_{-M1,-M2+1} & \cdots & a_{-M1,-1} & a_{-M1,0} & a_{-M1,1} & \cdots & a_{-M1,M2-1} & a_{-M1,M2} \\ a_{-M1+1,-M2} & a_{-M1+1,-M2+1} & \cdots & a_{-M1+1,-1} & a_{-M1+1,0} & a_{-M1+1,1} & \cdots & a_{-M1+1,M2-1} & a_{-M1+1,M2} \\ \vdots & & \text{III} & & & & \text{IV} & & \vdots \\ a_{-1,-M2} & a_{-1,-M2+1} & \cdots & a_{-1,-1} & a_{-1,0} & a_{-1,1} & \cdots & a_{-1,M2-1} & a_{-1,M2} \\ a_{0,-M2} & a_{0,-M2+1} & \cdots & a_{0,-1} & a_{0,0} & a_{0,1} & \cdots & a_{0,M2-1} & a_{0,M2} \\ a_{1,-M2} & a_{1,-M2+1} & \cdots & a_{1,-1} & a_{1,0} & a_{1,1} & \cdots & a_{1,M2-1} & a_{1,M2} \\ \vdots & & \text{II} & & & & \text{I} & & \vdots \\ a_{M1-1,-M2} & a_{M1-1,-M2+1} & \cdots & a_{M1-1,-1} & a_{M1-1,0} & a_{M1-1,1} & \cdots & a_{M1-1,M2-1} & a_{M1-1,M2} \\ a_{M1,-M2} & a_{M1,-M2+1} & \cdots & a_{M1,-1} & a_{M1,0} & a_{M1,1} & \cdots & a_{M1,M2-1} & a_{M1,M2} \end{bmatrix}$$

Form 2a: M_1, M_2 even

$a_{\pm M1,\pm M2}$	$a_{\pm M1,-M2+1}$	...	$a_{\pm M1,-1}$		$a_{\pm M1,0}$	$a_{\pm M1,1}$	...	$a_{\pm M1,M2-1}$
$a_{-M1+1,\pm M2}$	$a_{-M1+1,-M2+1}$	...	$a_{-M1+1,-1}$		$a_{-M1+1,0}$	$a_{-M1+1,1}$	...	$a_{-M1+1,M2-1}$
.		III				IV		.
.								.
.								.
$a_{-1,\pm M2}$	$a_{-1,-M2+1}$	...	$a_{-1,-1}$		$a_{-1,0}$	$a_{-1,1}$	...	$a_{-1,M2-1}$
$a_{0,\pm M2}$	$a_{0,-M2+1}$	...	$a_{0,-1}$		$a_{0,0}$	$a_{0,1}$	...	$a_{0,M2-1}$
$a_{1,\pm M2}$	$a_{1,-M2+1}$	...	$a_{1,-1}$		$a_{1,0}$	$a_{1,1}$	...	$a_{1,M2-1}$
.		II				I		.
.								.
.								.
$a_{M1-1,\pm M2}$	$a_{M1-1,-M2+1}$	...	$a_{M1-1,-1}$		$a_{M1-1,0}$	$a_{M1-1,1}$	...	$a_{M1-1,M2-1}$

Form 3a: M_1, M_2 odd

$a_{0,0}$	$a_{0,1}$	...	$a_{0,M2-1}$	$a_{0,M2}$		$a_{0,-M2}$	$a_{0,-M2+1}$	...	$a_{0,-1}$
$a_{1,0}$	$a_{1,1}$	...	$a_{1,M2-1}$	$a_{1,M2}$		$a_{1,-M2}$	$a_{1,-M2+1}$	...	$a_{1,-1}$
.			I				II		.
.									.
.									.
$a_{M1-1,0}$	$a_{M1-1,1}$	...	$a_{M1-1,M2-1}$	$a_{M1-1,M2}$		$a_{M1-1,-M2}$	$a_{M1-1,-M2+1}$	...	$a_{M1-1,-1}$
$a_{M1,0}$	$a_{M1,1}$	...	$a_{M1,M2-1}$	$a_{M1,M2}$		$a_{M1,-M2}$	$a_{M1,-M2+1}$	...	$a_{M1,-1}$
$a_{-M1,0}$	$a_{-M1,1}$	...	$a_{-M1,M2-1}$	$a_{-M1,M2}$		$a_{-M1,-M2}$	$a_{-M1,-M2+1}$	...	$a_{-M1,-1}$
$a_{-M1+1,0}$	$a_{-M1+1,1}$	...	$a_{-M1+1,M2-1}$	$a_{-M1+1,M2}$		$a_{-M1+1,-M2}$	$a_{-M1+1,-M2+1}$	...	$a_{-M1+1,-1}$
.			IV				III		.
.									.
.									.
$a_{-1,0}$	$a_{-1,1}$	...	$a_{-1,M2-1}$	$a_{-1,M2}$		$a_{-1,-M2}$	$a_{-1,-M2+1}$	...	$a_{-1,-1}$

Form 4a: M_1, M_2 even

$a_{0,0}$ $a_{0,1}$ ... $a_{0,M2-1}$	$a_{0,\pm M2}$ $a_{0,-M2+1}$ ... $a_{0,-1}$
$a_{1,0}$ $a_{1,1}$ ... $a_{1,M2-1}$	$a_{1,\pm M2}$ $a_{1,-M2+1}$ ... $a_{1,-1}$
I	II
$a_{M1-1,0}$ $a_{M1-1,1}$ ... $a_{M1-1,M2-1}$	$a_{M1-1,\pm M2}$ $a_{M1-1,-M2+1}$ ... $a_{M1-1,-1}$
$a_{\pm M1,0}$ $a_{\pm M1,1}$ ... $a_{\pm M1,M2-1}$	$a_{\pm M1,\pm M2}$ $a_{\pm M1,-M2+1}$ ... $a_{\pm M1,-1}$
$a_{-M1+1,0}$ $a_{-M1+1,1}$ ... $a_{-M1+1,M2-1}$	$a_{-M1+1,\pm M2}$ $a_{-M1+1,-M2+1}$ ... $a_{-M1+1,-1}$
IV	III
$a_{-1,0}$ $a_{-1,1}$ ... $a_{-1,M2-1}$	$a_{-1,\pm M2}$ $a_{-1,-M2+1}$ ... $a_{-1,-1}$

with $a_{k,\pm M2} = a_{k,M2} + a_{k,-M2}$ and $a_{\pm M1,l} = a_{M1,l} + a_{-M1,l}$

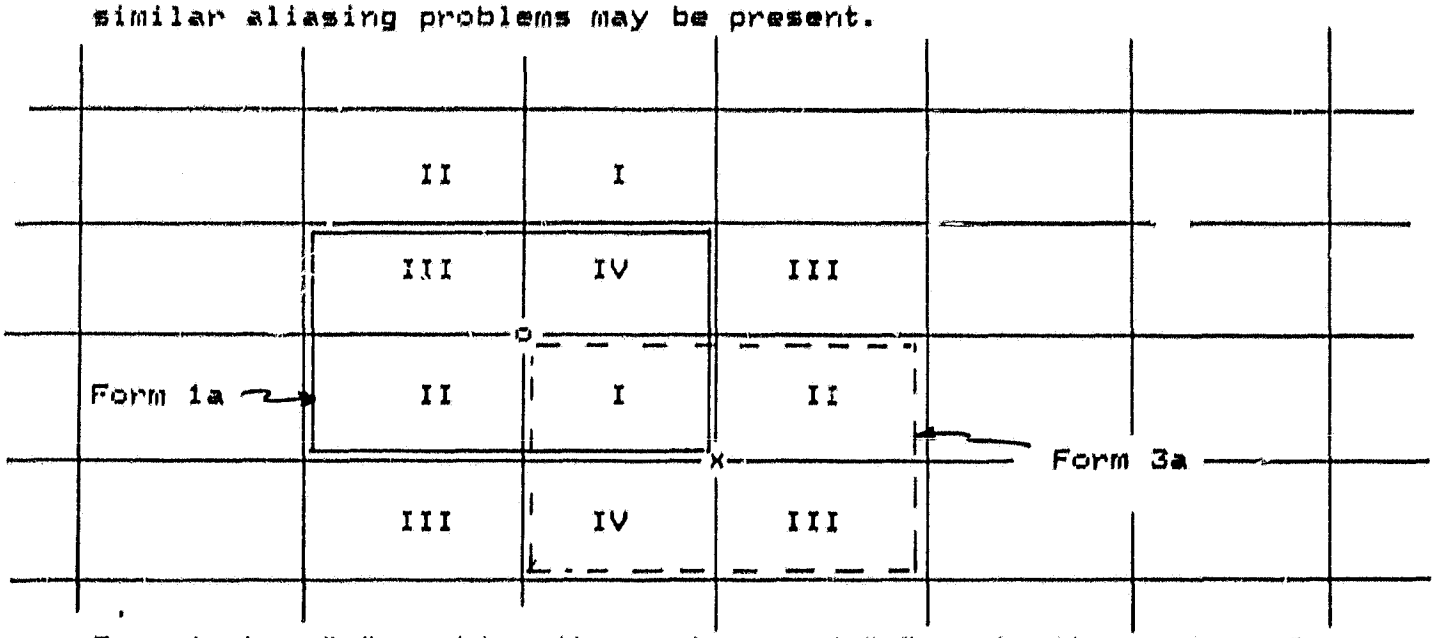
and $a_{\pm M1,\pm M2} = a_{M1,M2} + a_{-M1,M2} + a_{M1,-M2} + a_{-M1,-M2}$.

The quadrant numbers are shown as Roman numerals for Forms 1a, 2a, 3a, and 4a. The quadrant numbering system depends upon the choice of the x-y coordinate system discussed above. Elements $a_{0,i}$ and $a_{i,0}$ are <u>on</u> the axes. They are <u>between</u> quadrants and are not actually <u>in</u> them. The rearranged sections from Form 1a to Form 3a are not all of the same size; they do not have the same numbers of columns and rows. Note that for N even, Form 2a, the origin is not at the geometric center of the matrix, but is just to the right of and just below the center.

One may rearrange Form 1a into Form 3a or Form 2a into Form 4a and then take the standard DFT. The result will represent the data with the correct origin of Form 1a or Form 2a. Alternatively, one may take the transform of Form 1a as it

is for N_1, N_2 odd (or Form 2a for N_1, N_2 even), and then phase multiply the transform to have it correctly represent the origin as in Form 1a or Form 2a.

In one dimension, the data are replicated and aliasing may occur. Two-dimensional data are also replicated and similar aliasing problems may be present.



Form 1a has "o" marking its center, and "x" marks the center of Form 3a.

The phase multiplication in the transform domain to get the transform of data with the origin in the center will be

$$\exp(+2\pi i u M_2 \Delta x)\ \exp(+2\pi i v M_1 \Delta y) \quad ,$$

where the uv origin is in the center or the upper left corner of the matrix, depending on whether the transform has been rearranged or not. Since for N_1 and N_2 even, $M_2 = N_2/2$, $M_1 = N_1/2$, and since $\Delta u = 1/(N_2 \Delta x)$, and $\Delta v = 1/(N_1 \Delta y)$, then the exponentials above become

$$\exp(+2\pi i (N_2 \Delta x/2)(s/N_2 \Delta x))\ \exp(+2\pi i (N_1 \Delta y/2)(r/N_1 \Delta y))$$

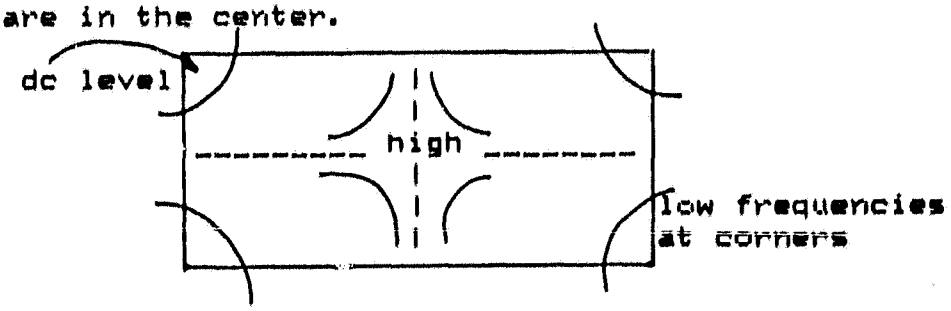
$$= \exp(i\pi s)\ \exp(i\pi r) = (-1)^s\ (-1)^r = (-1)^{r+s} \quad .$$

where r and s should be measured from the origin.

The normal way to think of the transform is Form 1A, which may be obtained from the diagram for Form 1a by substituting capital letter A for small a. However, the standard DFT calculated is Form 3A for N odd and Form 4A for N even.

For Form 4A, the dc level, $A_{0,0}$, is in the upper left corner, and lower frequencies, i.e., those of small |u|, |v| values, are found in the corners of the matrix. The highest frequencies are in the center.

Form 4A



Therefore, for two-dimensional low-pass filters, the center region must be blocked out. Note that the filter covers a two-dimensional area.

If the transform origin is to be in the "center", rearrange the transform after taking it, or phase multiply the data before taking the transform:

$$\exp(-2\pi i M_2 \Delta u m \Delta x)\ \exp(-2\pi i M_1 \Delta v n \Delta y)$$

$$= \exp(-2\pi i N_2 m \Delta x/(2N_2\Delta x))\ \exp(-2\pi i N_1 n \Delta y/(2N_1\Delta y))$$

$$= (-1)^{m+n} \qquad \text{for } N_1,\ N_2 \text{ even} \quad .$$

The transform will be in Form 1A for N_1, N_2 odd, or Form 2A for N_1, N_2 even, for either method.

With currently defined FFT computer algorithms, the initial data are assumed to be in Form 0a or Form 3a for N_1, N_2 odd and Form 0a or Form 4a for N_1, N_2 even. The distinction

between Forms 0a and 3a or 4a does not matter if the data are never considered as giving the "frequency content" of the forward transform.

ONE-DIMENSIONAL REAL DATA

Whether calculating the transform of two real arrays with one complex transform, or the transform of one N-point real array with one (N/2)-point complex transform (Brigham, 1974), or the ordinary complex transform for real data, computer storage requirements can be reduced by making use of the symmetries of the transform for real data. The function domain forms do not change. The transform domain forms can be modified as follows:

Form 1A:

A_{-M} A_{-M+1} ... A_{-2} A_{-1} A_0

for N odd, with ((N-1)/2) + 1 complex terms. It seems that there are 2[((N-1)/2) + 1] = N + 1 independent pieces of data, which is one too many; but A_0 is real, so that there are actually only N independent values as there should be. To fill in the rest of the terms, make use of the Hermitian property:

$$A_i = A^*_{-i} .$$

Form 3A:

A_0 A_1 A_2 ... A_{M-1} A_M

for N odd. There are N numbers represented as in Form 1A. Again use the Hermitian property to fill in the missing terms:

$$A_{-i} = A^*_i .$$

Form 4A:

$A_0 \quad A_1 \quad A_2 \ldots A_{M-1} \quad A_{M,-M}$

for N even, where both A_0 and $A_{M,-M}$ are real. (This again insures that there are $2[(N/2) + 1] - 2 = N$ independent pieces of information.)

Form 2A:

$A_{M,-M} \quad A_{-M+1} \ldots A_{-2} \quad A_{-1} \quad A_0$.

For real data one may choose to work only with positive frequencies, Form 3A or Form 4A.

CALCULATION OF THE TWO-DIMENSIONAL DFT

Recall that the definition of the two-dimensional DFT is

$$A_{rs} = \sum_{k=0}^{N_1-1} \sum_{\ell=0}^{N_2-1} a_{k\ell} \exp\{-i2\pi[(rk/N_1)+(s\ell/N_2)]\}$$

$$= \sum_{k=0}^{N_1-1} \exp(-i2\pi rk/N_1) \sum_{\ell=0}^{N_2-1} a_{k\ell} \exp(-i2\pi s\ell/N_2) \quad ,$$

where k varies vertically and ℓ varies horizontally. The inner summation over ℓ is the sum over the k^{th} row for each k. The rule implied in this expression is that row transforms are taken first and then the column transforms. Let

$$\alpha_{ks} = \sum_{\ell=0}^{N_2-1} a_{k\ell} \exp(-i2\pi s\ell/N_2) \quad .$$

Then the summation over k becomes

$$\sum_{k=0}^{N_1-1} \alpha_{ks} \exp(-i2\pi rk/N_1) \quad .$$

This is a sum over the s^{th} column for each s. The rule implied by this expression is that the column transform is next taken. Therefore the complete rule for the two-dimensional DFT is to

take all the row transforms and then all the column transforms of this result. Bookkeeping may be a problem! Furthermore, the column transforms may alternatively be taken first and then the row transforms of that result.

MATRIX APPROACH

Following the approach of Brigham (1974), the Fourier transform [A] of original data [a] may be written as

$$\begin{pmatrix} A_0 \\ A_1 \\ A_2 \\ A_3 \end{pmatrix} = \begin{pmatrix} W^0 & W^0 & W^0 & W^0 \\ W^0 & W^1 & W^2 & W^3 \\ W^0 & W^2 & W^4 & W^6 \\ W^0 & W^3 & W^6 & W^9 \end{pmatrix} \begin{pmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \end{pmatrix} ,$$

with $W = \exp(i2\pi/N)$. This matrix equation describes the transformation of one column vector (a_i) into another column vector (A_i). The [W] matrix performs the column transform by pre-multiplying the column vector (a_i). If the [W] matrix premultiplies a matrix [a], it will transform each column of that matrix. The corresponding transform operator matrix which would give a row transform is $[W^T]$, which post-multiplies a row vector $(a_i)^T = (a_0, a_1, a_2, a_3)$. If $[W^T]$ post-multiplies the matrix [a], it transforms every row of [a]. Therefore for a two-dimensional matrix transformation of the matrix [a],

$$[W]\,[a]\,[W^T] = [A] \quad .$$

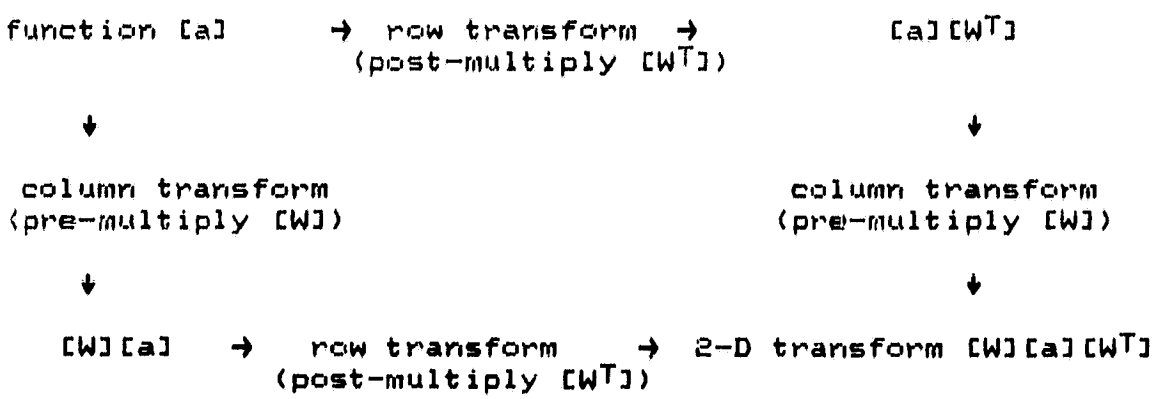
This shows that either row transforms or column transforms could be calculated first, because matrix multiplication is associative:

$$([W]\,[a])\,[W^T] = [W]\,([a]\,[W^T]) = [A] \quad .$$

For a square matrix, the dimensions of $[W^T]$ are the same

as those for [W] and $[W^T] = [W]$. If the matrix [a] is not square, say N_1 rows and N_2 columns, then the pre-multiplying matrix [W] will be N_1 by N_1 and the post-multiplying matrix $[W^T]$ will be N_2 by N_2.

The following diagram illustrates the two equivalent paths which may be used to obtain the two-dimensional Fourier transform.



TWO-DIMENSIONAL REAL DATA

For real data, the transform of two rows at a time can be obtained using one complex transform (Brigham, 1974). Whether or not this is done, we begin by assuming the data are a matrix of size N_1 by N_2 arranged in Form 0a:

$$\begin{bmatrix} a_{0,0} & \cdots & a_{0,N2-1} \\ \vdots & & \vdots \\ a_{N1-1,0} & \cdots & a_{N1-1,N2-1} \end{bmatrix} .$$

Let α_{ij} represent the matrix elements for the intermediate step after taking row transforms of the original matrix [a]. Then the one-dimensional row transforms of the real rows of [a] produce

$$\begin{bmatrix} \alpha_{0,0} & \alpha_{0,1} \cdots & \alpha_{0,(N2/2)-1} & \alpha_{0,N2/2} & \alpha^*_{0,(N2/2)-1} \cdots & \alpha^*_{0,1} \\ \vdots & & & & \vdots & \\ \alpha_{N1-1,0} & \alpha_{N1-1,1} \cdots & \alpha_{N1-1,(N2/2)-1} & \alpha_{N1-1,N2/2} & \alpha^*_{N1-1,(N2/2)-1} \cdots & \alpha^*_{N1-1,1} \end{bmatrix}$$

The redundant part is the right side with the complex conjugates. It is written in terms of the non-redundant part. The column just to the left of the redundant part, column $N_2/2$, is called the boundary element column. Using $M_1 = N_1/2$ and $M_2 = N_2/2$, the α matrix may be relabeled as

$$\begin{bmatrix} \alpha_{0,0} & \cdots & \alpha_{0,M2-1} & \alpha_{0,\pm M2} & \alpha^*_{0,M2-1} & \cdots & \alpha^*_{0,1} \\ \vdots & & & & & & \\ \alpha_{M1-1,0} \cdots & \alpha_{M1-1,M2-1} & \alpha_{M1-1,\pm M2} & \alpha^*_{M1-1,M2-1} & \cdots & \alpha^*_{M1-1,1} \\ \alpha_{\pm M1,0} & \cdots & \alpha_{\pm M1,M2-1} & \alpha_{\pm M1,\pm M2} & \alpha^*_{\pm M1,M2-1} & \cdots & \alpha^*_{\pm M1,1} \\ \alpha_{-M1+1,0} \cdots & \alpha_{-M1+1,M2-1} & \alpha_{-M1+1,\pm M2} & \alpha^*_{-M1+1,M2-1} & \cdots & \alpha^*_{-M1+1,1} \\ \vdots & & & & & & \\ \alpha_{-1,0} & \cdots & \alpha_{-1,M2-1} & \alpha_{-1,\pm M2} & \alpha^*_{-1,M2-1} & \cdots & \alpha^*_{-1,1} \end{bmatrix}$$

Elements of the first column and the boundary element column are real because the original data were real.

Next the column transforms are taken to obtain

Form 4A:

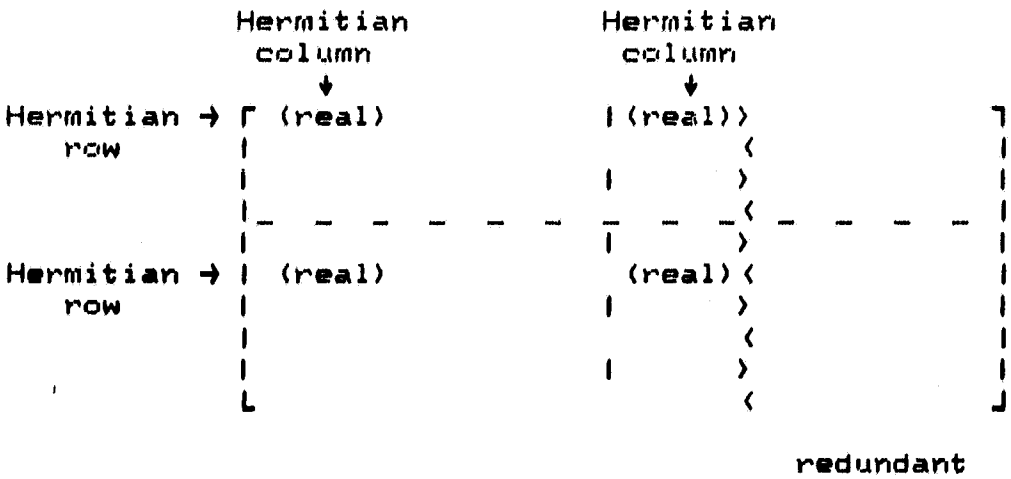
$$
\begin{bmatrix}
A_{0,0} & A_{0,1} & \cdots & A_{0,M2-1} & A_{0,\pm M2} & A^*_{0,M2-1} & \cdots & A^*_{0,1} \\
 & & & & & A^*_{-1,M2-1} & & A^*_{-1,1} \\
 & & & & & & & \\
 & & I & & & & II & \\
 & & & & & & & \\
 & & & & & A^*_{-M1+1,M2-1} & \cdots & A^*_{-M1+1,1} \\
A_{M1-1,0} & A_{M1-1,1} & \cdots & A_{M1-1,M2-1} & A_{M1-1,\pm M2} & & & \\
A_{\pm M1,0} & A_{\pm M1,1} & \cdots & A_{\pm M1,M2-1} & A_{\pm M1,\pm M2} & A^*_{\pm M1,M2-1} & \cdots & A^*_{\pm M1,1} \\
 & & & & & A^*_{M1-1,M2-1} & & A^*_{M1-1,1} \\
 & & & & & & & \\
 & & IV & & & & III & \\
 & & & & & & & \\
 & & & & & A^*_{1,M2-1} & \cdots & A^*_{1,1} \\
A_{-1,0} & A_{-1,1} & \cdots & A_{-1,M2-1} & A_{-1,\pm M2} & & &
\end{bmatrix}
$$

The first column and the boundary element column are both Hermitian. The redundant part is written in terms of the complex conjugates of the non-redundant part. (Note that the low-frequency edges of the quadrants are not at the center but at the far corners.) The first subscript of the redundant part of a row is usually not the same as that of the non-redundant part. The explanation is given as follows. In one dimension the continuous Fourier transform of $f^*(x)$ is $F^*(-s)$, and in two dimensions the discrete Fourier transform of $a^*_{j,k}$ is $A^*_{-j,k}$.

Except for the first row, quadrant II (III) of the [A] matrix above comes from conjugating and rearranging part of quadrant IV (I). This symmetry can be seen readily in photographs of two-dimensional optical transforms, for example, in Andrews (1970), pages 34, 35, 42, 107-109, 165-167. In these photographs, the logarithms of the Fourier transforms are shown in order to decrease the dynamic range of the transform and allow the small values to be seen with the limited dynamic

range of an image display.

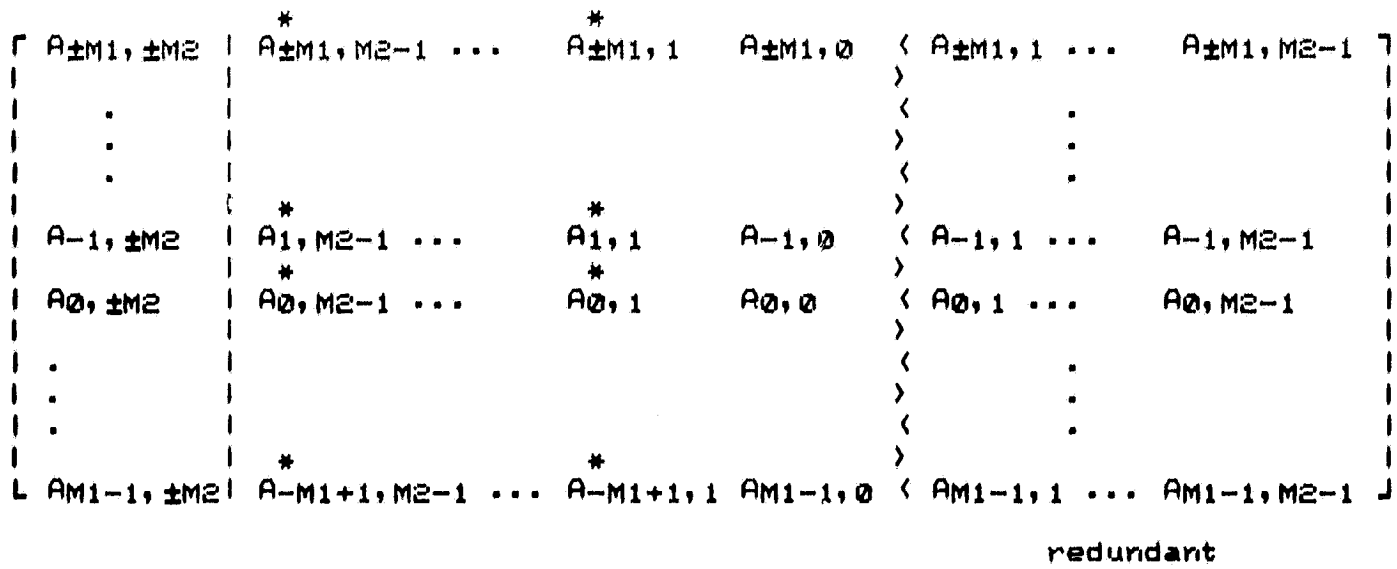
Recall that either the row transforms or the column transforms can be taken first for the two-dimensional Fourier transform. If the column transforms are obtained first, two real rows (in contrast to columns) are obtained. When next the row transforms are taken, two Hermitian rows (in contrast to columns) are obtained. The final transform matrix [A] is the same as that obtained by taking row transforms first and then column transforms, and it may therefore be written as



The real elements are indicated. Although the right "half" is shown to be redundant, the bottom "half" may be taken as the redundant part, as would be natural after taking column transforms.

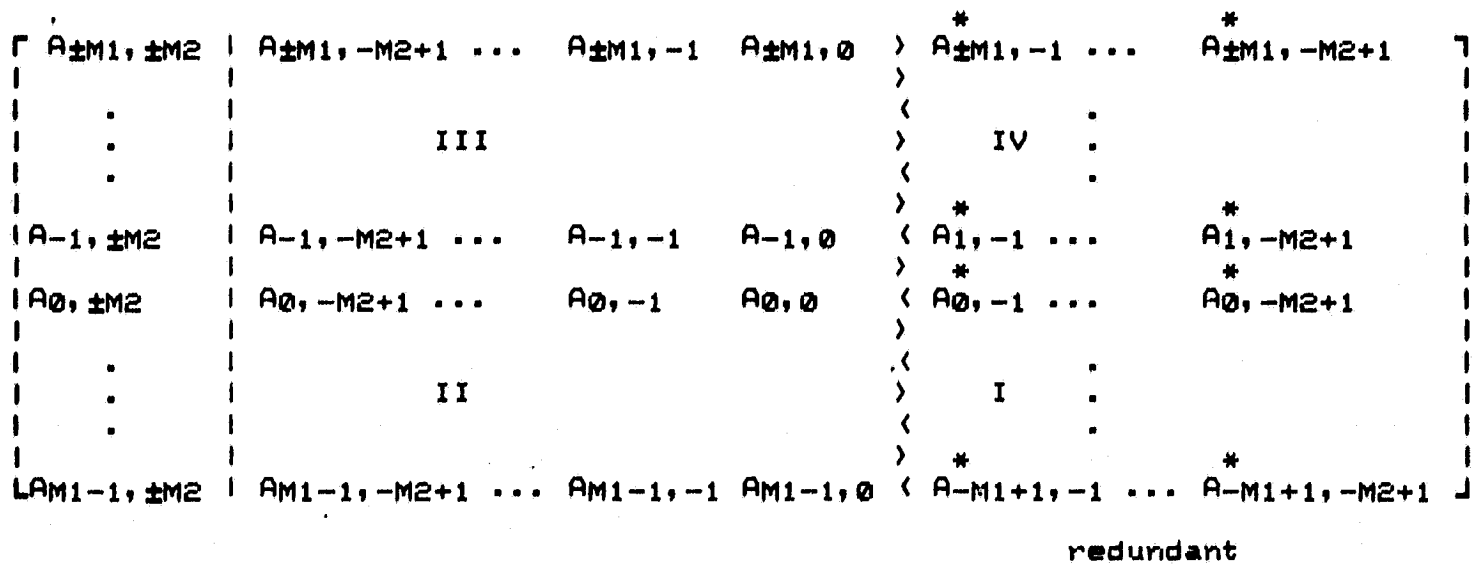
Form 4A can be rearranged into Form 2A, either by function domain phase multiplication as previously defined, or equivalently by transform domain rearrangement. Transform domain rearrangement gives

Form 2A:



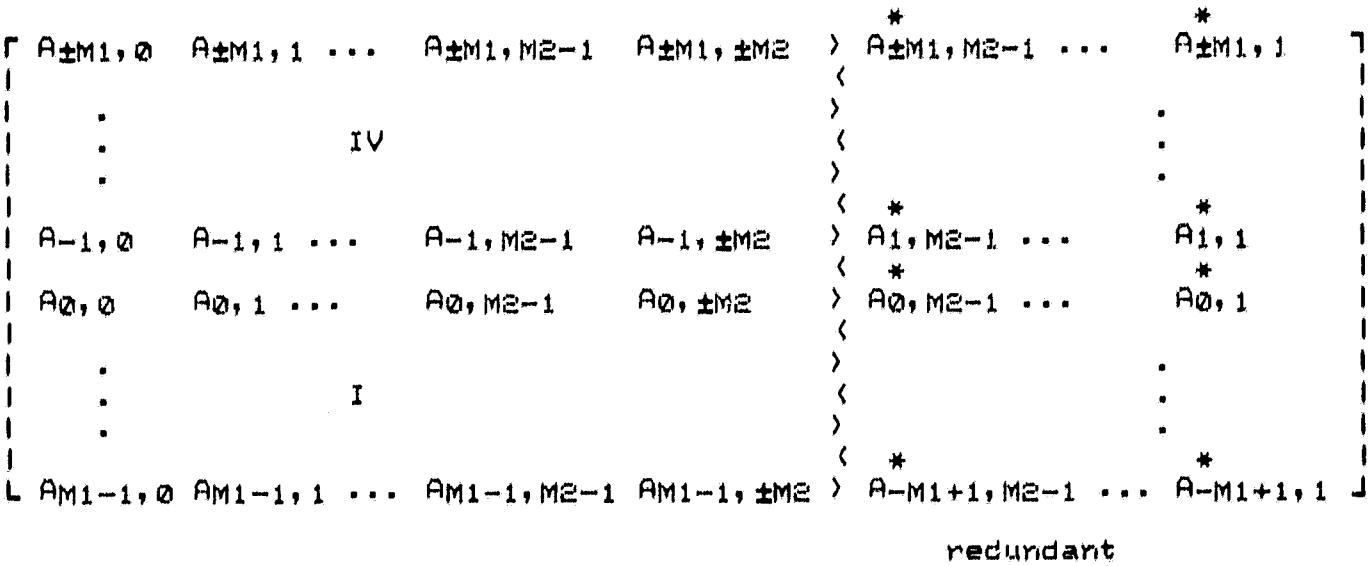
The dashed line is the redundant line from Form 4A. Form 2A should be rewritten in terms of known values, so that the redundant part is given in terms of the non-redundant part.

Form 2A:

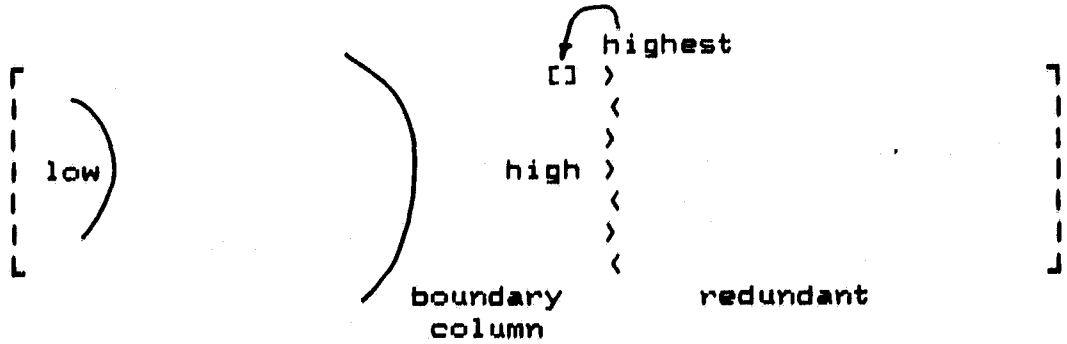


If positive quadrants are desired rather than negative for the non-redundant part, a phase shift should be performed only on y for the data (or rearrangement on v in Form 4A).

The resulting form is:



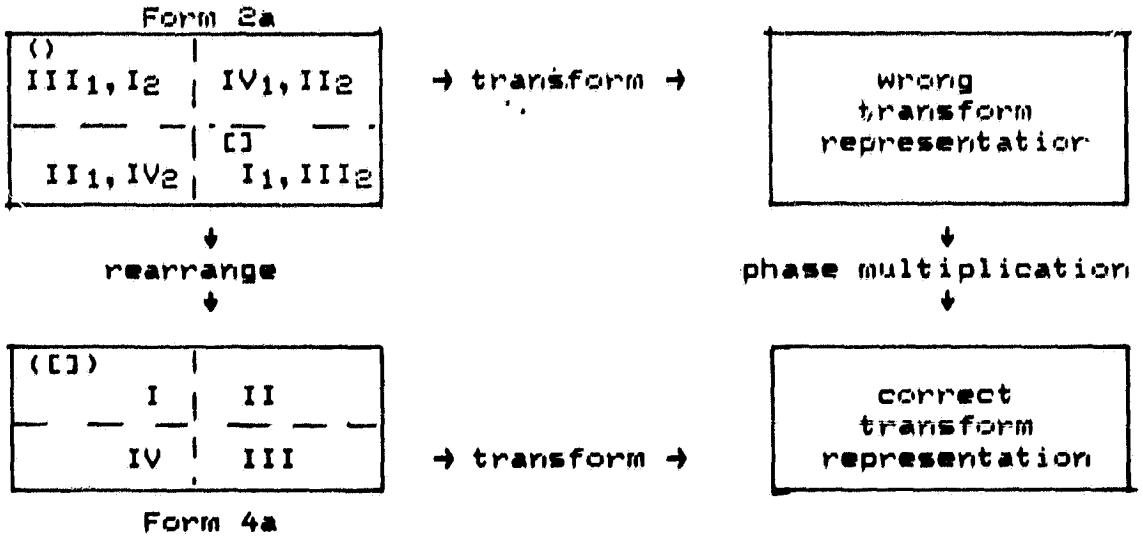
The required phase multiplication in the function domain is with $(-1)^{j}$, not $(-1)^{k+j}$. Only the distance from the x axis (the y measurement) is used. This puts the high frequencies together and the low frequencies together in the non-redundant part of the transform and keeps only positive frequencies. The boundary-element column contains the highest frequencies. The phase multiplication can be done either before or after taking the row transforms to get the α matrix.



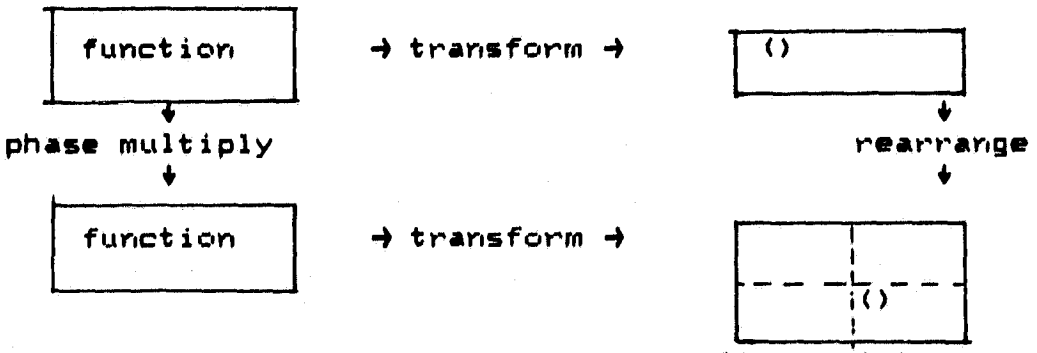
SUMMARY PICTURES

For complex two-dimensional data, there are two possible paths to the correct transform representation if the origin is

in the center. In the following diagram, the actual origin of the two-dimensional data is marked with [], the origin assumed by the FFT algorithm is marked with (), subscript 1 on quadrant numbers indicates the actual quadrants, and subscript 2 indicates the quadrant numbers as assumed by the FFT algorithm.



There are also two possible paths for obtaining the transform origin in the center:



It is possible that parts of both of these pictures will be applied to the data.

The notation of Oppenheim and Schaffer will be used. Consider two arrays: $x_1(m,n)$ of dimension $M_1 \times N_1$, and $x_2(m,n)$ of dimension $M_2 \times N_2$. The linear convolution may be defined as

$$x_3(m,n) = \sum_{q=-\infty}^{\infty} \sum_{r=-\infty}^{\infty} x_1(q,r)\, x_2((m-q),(n-r)) \quad .$$

The limits on the summations depend on whether the convolution is expanding or non-expanding.

It is possible to define for computational purposes either expanding or non-expanding convolutions. As an example in one dimension, consider two functions: f (of length n_f) and g (of length n_g). The convolution $h = f * g$ will have length $n_h = n_f + n_g - 1$, which is the normal expanding convolution result. For the non-expanding result, let f be the input, g the filter, and the output required to have the same length as the input, i.e., $n_h = n_f$. The expansion of f caused by convolution with g is dropped. To save computer time it would not be calculated at all. A non-expanding result is important in (1) iterative calculations, where the output would expand with each iteration, (2) two dimensions, when computer storage for only a certain size image is available, and (3) when one is interested in the filtered result only over the domain of the input. FFT programs might not conveniently accommodate the expanded arrays, especially in two dimensions. The limits of the summations in the linear convolution do not depend on the output variables for the expanding convolution; for non-expanding, they do. Note that the size of $x_3(m,n)$ defined above will be $(M_1+M_2-1) \times (N_1+N_2-1)$ for an expanding calculation.

PERIODIC CONVOLUTION

In one dimension, cyclic convolution was pictured with the two functions being convolved written on the circumference of cylinders which were rotated with respect to each other. The model for two dimensions could be one sphere inside another, with the spheres rotated with respect to each other.

The periodic convolution is defined to be

$$\tilde{x}_3(m,n) = \sum_{q=0}^{M-1} \sum_{r=0}^{N-1} \tilde{x}_1(q,r)\, \tilde{x}_2((m-q),(n-r)) \quad .$$

M and N must be specified along with the sum to completely define the periodic convolution, since the results of the periodic convolution will depend on the choice of M and N. For this output, $m = 0,1,\ldots,M-1$ and $n = 0,1,\ldots,N-1$. However, $\tilde{x}_3(m,n)$ need not be calculated in this order, i.e., from 0 to M-1 and 0 to N-1. It may be calculated in any order.

If $x_1(m,n)$ has dimension $M_1 \times N_1$ and $x_2(m,n)$ has dimension $M_2 \times N_2$, then the periodic convolution can be calculated for $M \geq M_1$ and $M \geq M_2$ and $N \geq N_1$ and $N \geq N_2$. The functions x_1 and x_2 must be packed with zeros to make them M x N periodic. Therefore x_1 and x_2 must be filled to the same size to define the periodic convolution. Then the result of the periodic convolution will correspond to that obtained from the DFT Convolution Theorem, using the DFT's of x_1 and x_2, each of size M x N. x_1 and x_2 must be packed with zeros to this size. Then

$$\mathcal{F}\{\tilde{x}_3(m,n)\} = \tilde{X}_1(k,\ell)\, \tilde{X}_2(k,\ell) \quad .$$

For the periodic result to agree with the linear, $M \geq M_1 + M_2 -$

References

H. C. Andrews, 1970, "Computer Techniques in Image Processing," Academic Press

Ronald N. Bracewell, 1978, "The Fourier Transform and its Applications," McGraw-Hill

E. Oran Brigham, 1974, "The Fast Fourier Transform," Prentice-Hall

A. V. Oppenheim and R. W. Schafer, 1975, "Digital Signal Processing," Prentice-Hall